

Instrukcji kompilacji i wykrozystania Simple Ontology Visualization API

Spis treści:

1. Opis projektu
 - 1.1 Opis modułu biblioteki SOVA
 - 1.2 Opis modułu wtyczki do programu Protege
 - 1.3 Opis modułu demo
 - 1.3.1 Przygotowanie modułu demo przed budowaniem
 - 1.3.2 Uruchomienie niezależnej aplikacji demonstracyjnej
2. Budowanie projektu
 - 2.1 Instalacja niezbędnych bibliotek
 - 2.2 Budowanie wszystkich modułów jednocześnie
 - 2.3 Budowa modułów pojedynczo
3. Zmiana numeru wersji projektu

1 Opis projektu

Projekt SOVA, stworzony z Mavenie, składa się z 3 modułów:

- biblioteka SOVA (w katalogu: SOVA),
- wtyczka do programu Protege (w katalogu: pluginProtegeSova),
- demo – samodzielna aplikacja do uruchomienia biblioteki SOVA, bez potrzeby instalacji pluginu w Protege (w katalogu: demo).

Projekt stworzony w Mavenie umożliwia zbudowanie wszystkich modułów jednocześnie lub pojedynczo.

1.1 Opis modułu biblioteki SOVA

Moduł biblioteki SOVA stanowi główną część projektu i służy do wizualizacji grafów.

1.2 Opis modułu wtyczki do programu Protege

Moduł wtyczki do programu Protege tworzy plik formatu .jar, który umożliwia wizualizację grafów w programie Protege. Wtyczkę należy umieścić w katalogu *plugins* programu Protege, żeby móc później ją uruchomić w tym programie.

1.3 Opis modułu demo

Moduł aplikacji demonstracyjnej służy do demonstracji funkcjonowania wtyczki, bez konieczności instalacji wtyczki w programie Protege. Moduł tworzy plik formatu .jar, który jest samodzielną aplikacją.

1.3.1 Przygotowanie modułu demo przed budowaniem

Moduł demo wymaga odpowiedniego przygotowania aplikacji przed użyciem.

Najpierw trzeba znaleźć plik Constants.java, który znajduje się w lokalizacji `root projektu\demo\src\main\java\org\pg\eti\kask\sova\demo`. Plik ten należy otworzyć dowolnym edytorem.

Następnie znajdź liniijkę:

```
public static final String ONTO_TEST_DIRECTORY = "X:\\Repozytoria\\sova\\doc\\OWL\\onot-test.owl";
```

Zadeklarowana tutaj wartość to adres lokalny do ontologii, która będzie wizualizowana w aplikacji demonstracyjnej. Należy ją ustawić odpowiednio dla siebie. Przykładowe wizualizacje można znaleźć w katalogu `root projektu\doc\OWL`. W tym miejscu konieczne jest podanie adresu bezwzględnego prowadzącego do pliku z ontologią.

Następnie trzeba znaleźć liniijkę:

```
public static final String PROPERTIES = "X:\\Repozytoria\\sova\\SOVA\\src\\visualization.properties";
```

W tym przypadku podany adres wskazuje na plik zawierający właściwości. W tym celu należy podać w tym miejscu adres bezwzględny znajdujący się w lokalizacji `root projektu\SOVA\src\visualization.properties`

Teraz wystarczy już tylko zbudować ten moduł.

1.3.2 Uruchomienie niezależnej aplikacji demonstracyjnej

W celu uruchomienia aplikacji demonstracyjnej trzeba otworzyć wiersz poleceń w katalogu `root projektu\demo\target`, a następnie wpisać komendę:

```
java -jar DEMO-0.8.5-jar-with-dependencies.jar
```

2. Budowanie projektu

2.1 Instalacja niezbędnych bibliotek

Przed pierwszym zbudowaniem trzeba zainstalować bibliotekę owlapi-bin. W tym celu trzeba otworzyć wiersz poleceń w katalogu lib, który znajduje się w korzeniu projektu (root projektu).

```
mvn install:install-file -Dfile=owlapi-bin.jar -DgroupId=owlapi-bin -DartifactId=owlapi-bin -Dversion=1 -Dpackaging=jar
```

To polecenie zainstaluje tę bibliotekę w lokalnym repozytorium MAVEN w celu późniejszego jego użycia przy budowaniu projektu.

2.2 Budowanie wszystkich modułów jednocześnie

W tym celu należy otworzyć katalog główny projektu (root projektu), a następnie uruchomić wiersz poleceń i wprowadzić polecenie:

```
mvn install
```

Wszystkie trzy moduły zostaną zbudowane w kolejności:

- biblioteka SOVA,
- wtyczka do programu Protege,
- demo - aplikacja demonstracyjna.

Plik w formacie .jar zostanie utworzony dla każdego z modułów w katalogu root/[katalog modułu]/target (moduły wraz nazwą katalogu w którym się znajdują są opisane w punkcie 1.).

2.3 Budowanie modułów pojedynczo

Każdy moduł może zostać zbudowany pojedynczo, bez konieczności budowania pozostałych. W tym celu należy otworzyć katalog zawierający dany moduł (moduły wraz nazwą katalogu w którym się znajdują są opisane w punkcie 1.). Następnie uruchomić wiersz poleceń i wprowadzić polecenie:

```
mvn install
```

Plik formatu .jar zostanie utworzony w katalogu `root projektu/[katalog modułu]/target`.

Uwaga!!! Moduły demo i wtyczka do programu Protege wymagają wcześniejszego zbudowania modułu SOVA. Po zbudowaniu modułu biblioteki SOVA, zostanie ona umieszczona w lokalnym repozytorium MAVEN, a pozostałe moduły będą z tego korzystać w trakcie budowy.

3. Zmiana numeru wersji projektu

W celu aktualizacji wersji projektu trzeba dokonać następujących zmian. Po pierwsze otworzyć plik pom.xml, który znajduje się w katalogu głównym całego projektu (root projektu) i znaleźć linię, która zawiera znacznik <version> oraz </version>. Wartość, która znajduje się między nimi jest numerem wersji projektu.

Następnie trzeba otworzyć katalog SOVA, który znajduje się w katalogu głównym całego projektu (root) i otworzyć plik pom.xml, który znajduje się w tym katalogu. Trzeba znaleźć następujący fragment:

```
<parent>
  <groupId>org.pg.eti.kask.ont.pluginSova</groupId>
  <artifactId>pluginSovaBuild</artifactId>
  <version>X.X.X</version>
  <relativePath>../pom.xml</relativePath>
</parent>
```

W znalezionym fragmencie trzeba zmodyfikować wartość, która znajduje się między znacznikami. Wprowadzona wartość musi być taka sama jak w pliku pom.xml, który został zmodyfikowany na początku.

W katalogach demo oraz pluginProtegeSova, które znajdują się w katalogu głównym całego projektu (root projektu), trzeba dokonać takich sam zmian jak w przypadku katalogu SOVA.